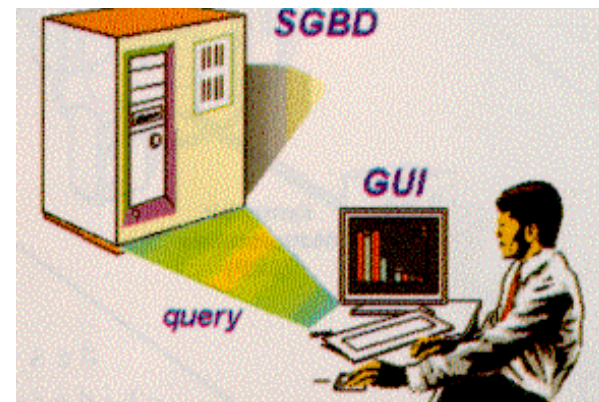


BAZE DE DATE



Ce este o BAZA DE DATE?

In sens larg un sistem proiectat pentru a oferi un mecanism organizat, capabil sa stocazeze, sa actualizeze si sa regaseasca informatia

Exemplu: o biblioteca

Noțiunea de bază de date este des vehiculată, prin ea înțelegându-se o mare cantitate de date, de informații, un mod de stocare electronică a datelor

Exemplu: cartea cu numerele de telefon, totalitatea foilor de observație dintr-un spital, carnetele de note ale elevilor

O bază de date nu este altceva decât o colecție de date (informații) ce descriu mai multe subiecte (obiecte) similare

Ce este un SISTEM DE GESTIONARE A BAZELOR DE DATE?

Baza de date este o colecție organizată de informații

Baza de date pentru a putea fi actualizata, modificata, utilizata in obtinerea de rapoarte si statistici, se afla sub controlul unui program. Întreg acest ansamblu poartă numele de SGBD (Sistem de Gestionare a Bazelor de Date), sau în engleză DBMS (Date Base Management Sistem).

In practica prin baza de date se intelege de multe ori si softul aferent, iar prin SGBD un sistem integrat in care se pot construi si interoga baze de date

Administratorul bazei de date. - supraveghează atât asupra integrității și confidențialității datelor cât și asupra gestionării utilizatorilor finali. Practic administratorul continuă activitatea **proiectantului SGBD-ului**.

Proiectantului SGBD-ului este cel care stabilește **structura bazei de date** (adică ce însusiri să descrie fiecare obiect, subiect). Etapa este deosebit de importantă, de atenția cu care este parcursă depinzând succesul ulterior al aplicației.

Trecerea structurii bazei de date într-un format electronic pe un disc de calculator poartă numele de **generare a bazei de date**. Odată generată structura, proiectantul transferă SGBD-ul administratorului, care trece la supravegherea completării cu informație. Această operație este efectuată de unul sau mai mulți **operatori** și este, de cele mai multe ori este continuă, derulându-se în paralel cu **consultarea de bazei de date** (întocmirea de rapoarte și statistici, plecând de la datele deținute).

Baze de date **RELATIONALE**

- înmagazinează datele în tabele care se pot lega logic după valorile anumitor coloane
- relația dintre câmpuri realizează asocierea explicită
- este un model simetric, uniformitatea reprezentării datelor determină uniformitate în mulțimea operatorilor
- bazat pe teoria matematică a relațiilor, proiectarea bazelor de date poate fi tratată algoritmic
- utilizează un limbaj standardizat de interogare a bazei de date numit **SQL**

Structura unei baze de date relationale

Informatiile sunt organizate in tabele, fiecare linie descriind cate un subiect. Liniile nu trebuie sa fie identice! O linie poarta numele de **inregistrare**.

Descrierea fiecarei insusiri a subiectului constituie un **camp**. Informatia dintr-un camp nu trebuie sa poata fi descompusa! Pentru fiecare câmp se alege **un nume, un tip de date și o gamă de reprezentare**.

	cod_identific	nume	prenume	greutate	inaltime	data_nast	sex	afectiune
	1	Ionescu	Dan	60	180	1/15/60	B	HTA
	2	Ionescu	Ana	50	175	5/25/70	F	sanatos
	3	Ionescu	Ion	75	180	6/28/50	B	HTA
▶	4	Pop	Dana	55	170	8/12/72	F	sanatos
	5	Vasile	Dana	45	150	9/10/85	F	sanatos
	6	Vasile	Dana	44	130	11/11/90	F	sanatos

Totalitatea campurilor ce descriu un subiect alcatuiesc o inregistrare. Totalitatea inregistrarilor alcatuiesc baza de date

camp

Camp cheie primara

camp 1	camp 2	camp 3	camp (n-1)	camp n
---------------	---------------	---------------	-------------	-------------------	---------------

The diagram illustrates a multi-camp system. It consists of a series of horizontal bars, each representing a camp. The bars are stacked vertically, with the bottom-most bar labeled 'camp 1' and the top-most bar labeled 'camp n'. The bars are arranged such that they overlap, with each subsequent bar shifted slightly to the right. The bars are divided into segments by vertical lines. The segments are labeled with '1' and '2' in a repeating pattern. The bottom-most bar is labeled 'camp 1', 'camp 2', 'camp 3', '....', 'camp (n-1)', and 'camp n' from left to right. The other bars are labeled with '1' and '2' in a repeating pattern.

	ID	Date of birth	Consultant	Surname	
	1	12/03/45	Mr Smith	Johnson	
	2	03/10/55	Dr Jones	Purves	
A field	3	23/12/81	Dr Anderson	Bull	A data item
	4	30/05/70	Mr Smith	Brind	
	5	05/01/75	Dr Jones	Green	
A record					

A row = a record

A column = a field

A cell = a data item; a particular value in a field for a particular record.

Tipuri uzuale de date (in Microsoft Access):

Text – folosit pentru informații ce nu depășesc 255 de caractere (litere,cifre,semne speciale)

Memo – folosit pentru informații ce depășesc 255 de caractere (64KO; ex:descrierea evoluției unei boli)

Number – folosit pentru informații numerice:

Byte – numere întregi în intervalul 0-255

Integer - numere întregi în intervalul -32768-+32767

Long integer - numere întregi în intervalul –2miliarde-+2miliarde

Single - numere zecimale în precizie simplă (7 cifre)

Double - numere întregi în intervalul (14 cifre)

Date/Time – folosit pentru informații de tip dată sau oră

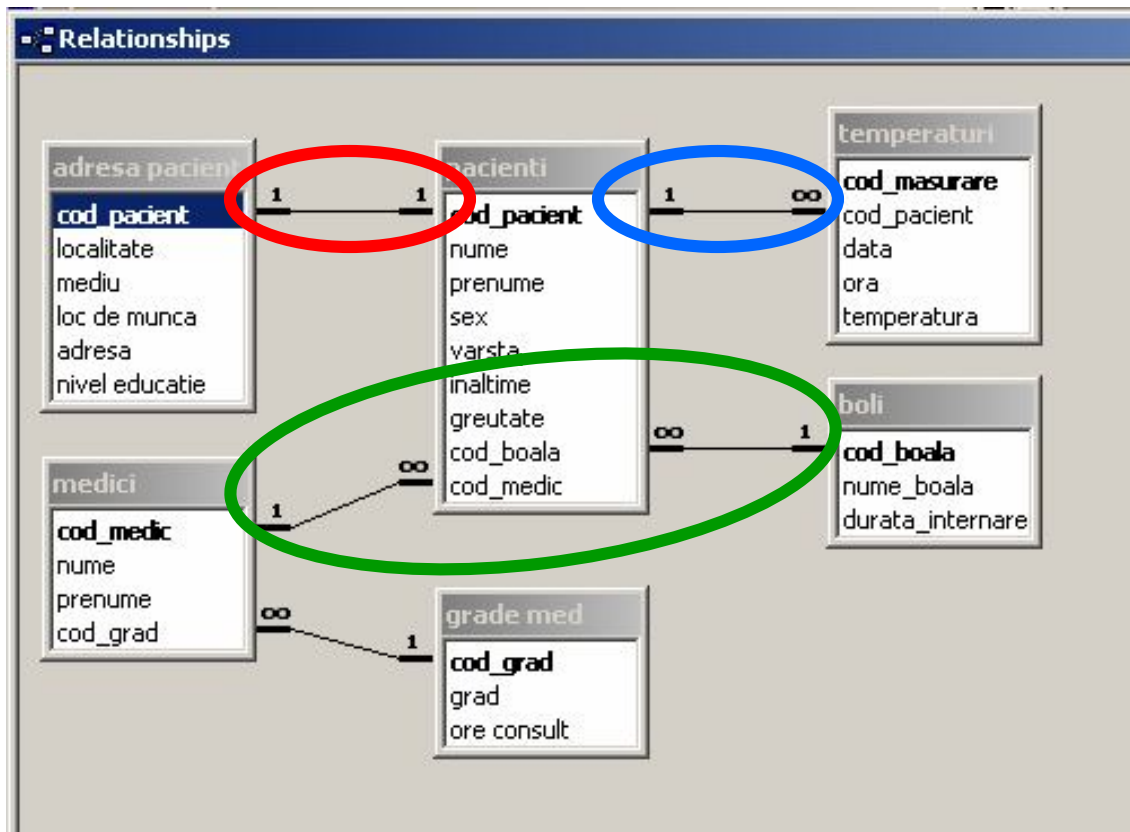
Currency – folosit pentru informații numerice în bani

Autonumber – folosit pentru generarea automată de numere în ordine crescătoare

Yes/No – folosit pentru informații binare de tip DA/NU

Proiectarea bazelor de date relationale

- identificarea entitatilor (tabelelor), relatiilor dintre ele si proprietatilor acestora**
- elaborarea unor scheme conceptuale ce sa ilustreze entitatile si legaturile dintre ele**
- normalizarea relatiilor (obtinerea de relatii cat mai simple, neincluse unele intre altele, eliminarea redundantelor, dar cu conservarea datelor si a dependentelor intre ele, in vederea eliminarilor confuziilor si greselilor)**

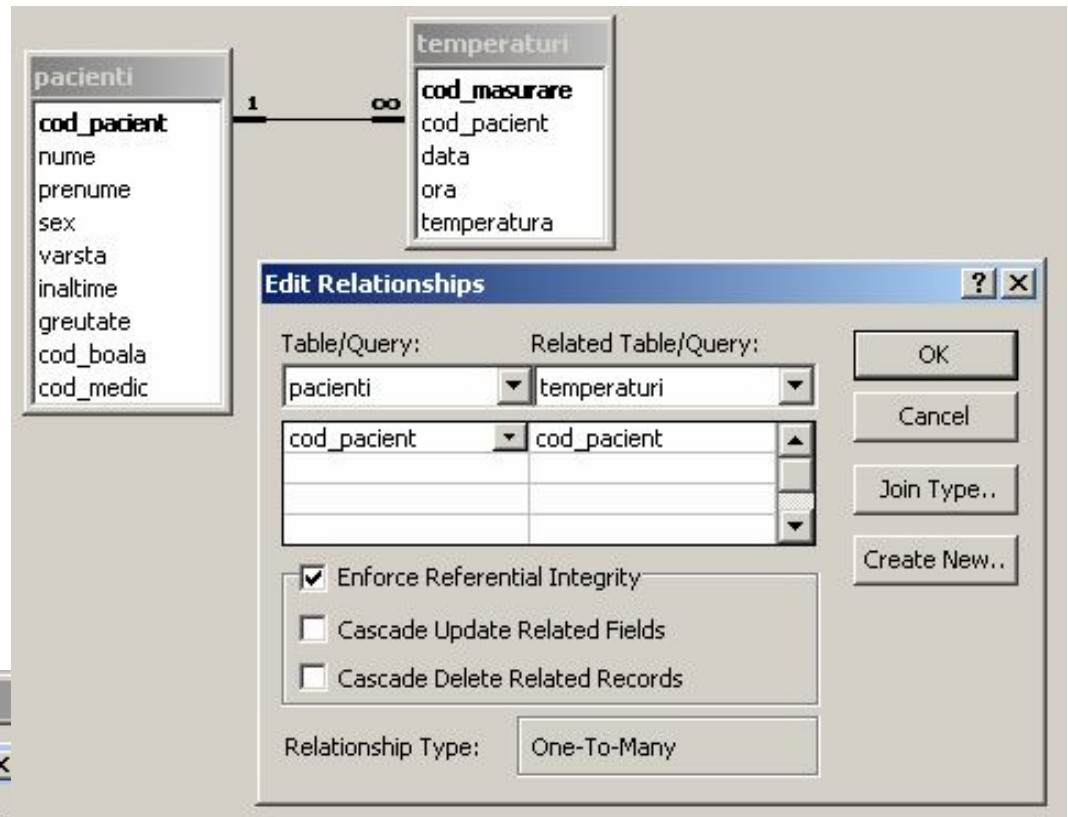
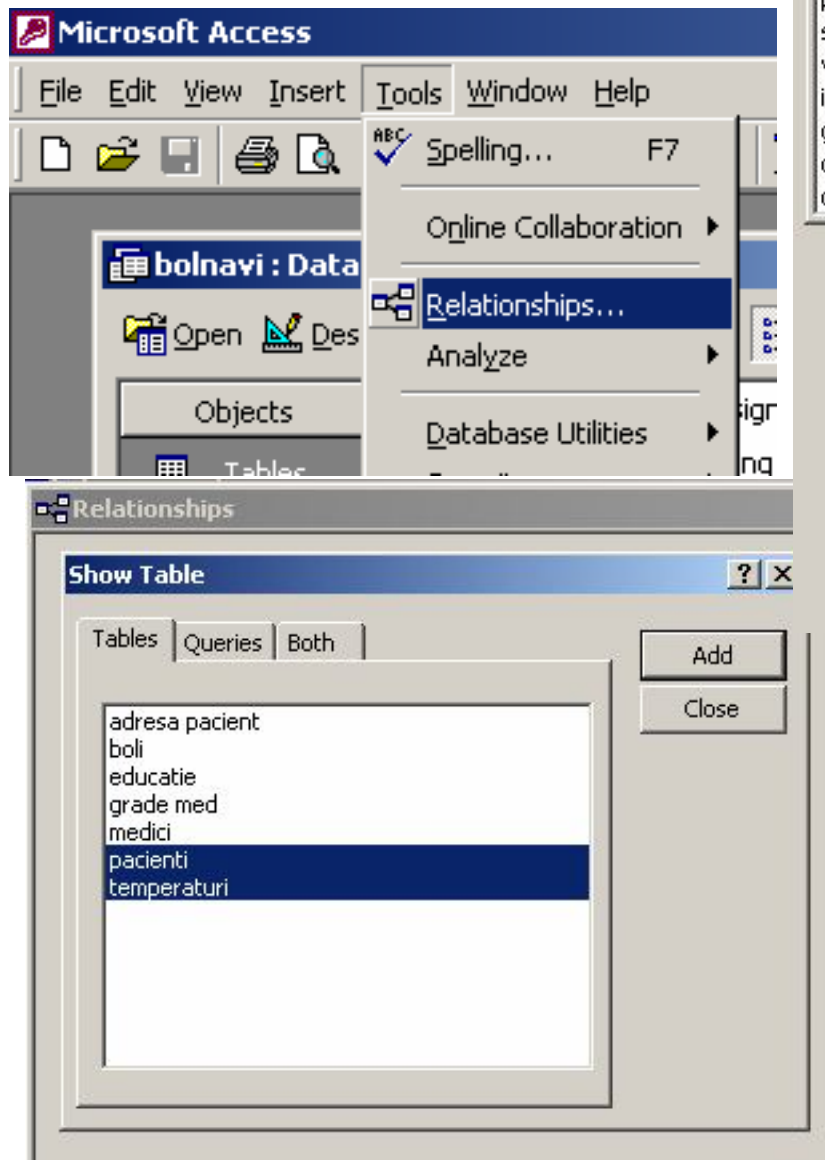


Relatii intre tabele

Campurile prin care se realizeaza legaturile trebuie sa fie de **acelasi tip**, in prima tabela fiind **cheie primara**, iar in a doua tot cheie primara sau **cheie straina** (adica primara in prima tabela)

- **1:1 (one to one)** fiecarei linii in primul tabel ii corespunde cel mult o singura linie in al doilea
- **1:m (one to many)** fiecarei linii in primul tabel ii pot corespunde mai multe in al doilea si fiecarei linii al doilea ii corespunde exact una in primul
- **m:m (many to many)** unei linii in primul tabel ii pot corespunde mai multe in al doilea si unei linii in al doilea tabel ii pot corespunde mai multe in primul

Fixarea legaturilor in Access



unei linii in primul tabel ii pot corespunde mai multe in al doilea si unei linii in al doilea tabel ii pot corespunde mai multe in primul

Normalizarea bazei de date

Informatiile nu trebuie repetate

Daca intr-un tabel dorim sa inseram coloane (campuri) existente deja in alt tabel, vom insera doar cheia primara a acelui tabel.

pacienti: Table

		cod_pacient	nume	prenume	sex	varsta	inaltime	greutate	cod_boala	cod_medic
▶	+	1	popescu	ion	m	80	170	70	1	3
	+	3	piru	alin	m	30	180	70	3	4
	+	4	oprescu	gigi	m	45	178	80	1	4
	+	5	vasilescu	bucur	m	15	160	60	5	5
	+	6	ispir	dana	f	10	150	30	2	5
	+	7	giga	ileana	f	8	110	30	5	1
*		(AutoNumber)								0

adresa pacient: Table

		cod_pacient	localitate	mediu	loc de munca	adresa	nivel educatie
▶	+	1	craiova	u	electroputere	calea bucuresti,	
	+	2	craiova	u	coca cola	1 mai, 5	
	+	3	bailesti	u	scoala generala	albastrele, 5	
	+	4	teasc	r	agricultor		
	+	5	pielesti	r	scola generala		
	+	6	filiasi	u	scoala generala	calea severinulu	
	+	7	craiova	u	scoala generala	craiovita, 77	
*		(AutoNumber)					

Alternativa in care in fiecare tabel repetam datele generale despre bolnav

pacienti : Table										
		cod_pacient	nume	prenume	sex	varsta	inaltime	greutate	cod_boala	cod_medic
▶	+	2	popescu	ion	m	80	170	70	1	3
	+	3	piru	alin	m	30	180	70	3	4
	+	4	oprescu	gigi	m	45	178	80	1	4
	+	5	vasilescu	bucur	m	15	160	60	5	5
	+	6	ispir	dana	f	10	150	30	2	5
	+	7	gica	ileana	f	8	110	30	5	1
*		(AutoNumber)				0	0	0	0	0

	cod_pacient	nume	prenume	sex	varsta	localitate	mediu	loc de munca	adresa
▶	2	popescu	ion	m	80	craiova	u	coca cola	1 mai, 5
	3	piru	alin	m	30	bailesti	u	scoala generala	albastrele, 5
	4	oprescu	gigi	m	45	teasc	r	agricultor	
	5	vasilescu	bucur	m	15	pielesti	r	scola generala	
	6	ispir	dana	f	10	filiasi	u	scoala generala	calea severinulu
	7	gica	ileana	f	8	craiova	u	scoala generala	craiovita, 77

Normalizarea bazei de date

Datele nu trebuie sa fie redundante

Tabelele trebuie construite in asa fel incat sa nu fie repetata informatia pe randuri.

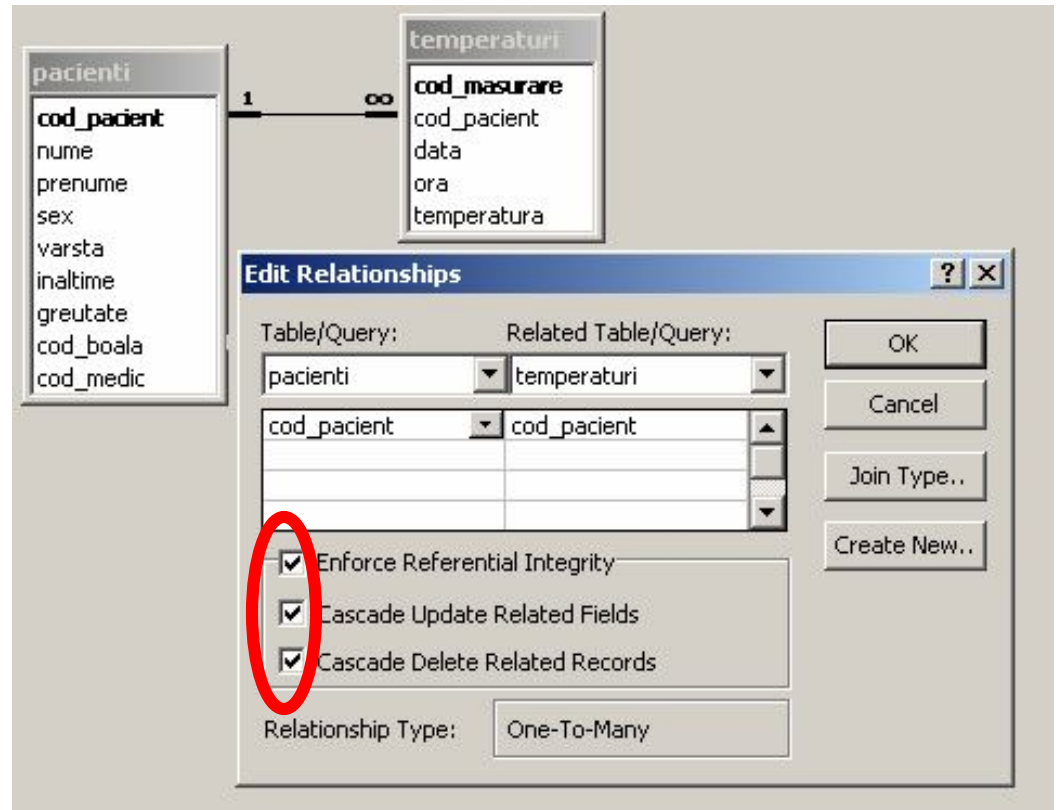
pacienti : Table							
	cod_pacient	nume	prenume	sex	varsta	inaltime	greutate
+	2	popescu	ion	m	80	170	70
+	3	piru	alin	m	30	180	70
+	4	oprescu	gigi	m	45	178	80
+	5	vasilescu	bucur		45	160	60
+	6	ispir	dana				
+	7	gica	ileana				
*	(AutoNumber)						

temperaturi : Table				
cod_masurare	cod_pacient	data	ora	temperatura
2	7	1/1/2006	8:00	38.5
3	7	1/1/2006	12:00	39
4	7	1/1/2006	16:00	39.5
5	7	1/1/2006	20:00	40
6	7	1/1/2006	0:00	39.5
7	7	1/2/2006	4:00	39.5
8	7	1/2/2006	8:00	39
9	2	1/2/2006	7:00	38
10	2	1/2/2006	15:00	38
11	2	1/2/2006	23:00	38.5
12	2	1/3/2006	7:00	38
13	2	1/3/2006	15:00	37.7
14	2	1/3/2006	23:00	37.5
15	2	1/4/2006	7:00	37.6
16	2	1/4/2006	15:00	37.5

Alternativa de tabel in care sunt inscrise toate temperaturile, la un loc cu datele generale despre pacient

	cod_pacient	nume	prenume	sex	varsta	inaltime	greutate	data	ora	temperatura
►	2	popescu	ion	m	80	170	70	1/2/2006	7:00	38
	2	popescu	ion	m	80	170	70	1/2/2006	15:00	38
	2	popescu	ion	m	80	170	70	1/2/2006	23:00	38.5
	2	popescu	ion	m	80	170	70	1/3/2006	7:00	38
	2	popescu	ion	m	80	170	70	1/3/2006	15:00	37.7
	2	popescu	ion	m	80	170	70	1/3/2006	23:00	37.5
	2	popescu	ion	m	80	170	70	1/4/2006	7:00	37.6
	2	popescu	ion	m	80	170	70	1/4/2006	15:00	37.5
	7	gica	ileana	f	8	110	30	1/1/2006	8:00	38.5
	7	gica	ileana	f	8	110	30	1/1/2006	12:00	39
	7	gica	ileana	f	8	110	30	1/1/2006	16:00	39.5
	7	gica	ileana	f	8	110	30	1/1/2006	20:00	40
	7	gica	ileana	f	8	110	30	1/1/2006	0:00	39.5
	7	gica	ileana	f	8	110	30	1/2/2006	4:00	39.5
	7	gica	ileana	f	8	110	30	1/2/2006	8:00	39

Reguli de integritate pentru baze de date relationale



- cheia primara a unui tabel trebuie sa fie unica si minimala
- pentru fiecare rand din tabel cheia primara trebuie sa aiba atribuita o valoare
- valoarea dintr-un tabel a unei **chei straine** trebuie sa se regaseasca intre valorile din tabelul unde este **cheie primara** (principiu valabil si la efectuarea de modificari in tabele)

Actualizarea informatiilor in Access in tabele cu legaturi

pacienti : Table								
cod_pacient	nume	prenume	sex	varsta	inaltime	greutate	cod_boala	cod_medic
2	popescu	ion	m	80	170	70	1	3
	cod_masurare	data	ora	temperatura				
	9	1/2/2006	7:00	38				
	10	1/2/2006	15:00	38				
	11	1/2/2006	23:00	38.5				
	12	1/3/2006	7:00	38				
	13	1/3/2006	15:00	37.7				
	14	1/3/2006	23:00	37.5				
	15	1/4/2006	7:00	37.6				
	16	1/4/2006	15:00	37.50				
*	(AutoNumber)			0				
3	piru	alin	m	30	180	70	3	4
4	oprescu	gigi	m	45	178	80	1	4
5	vasilescu	bucur	m	15	160	60	5	5
6	ispir	dana	f	10	150	30	2	5
7	gica	ileana	f	8	110	30	5	1
*	(AutoNumber)			0	0	0	0	0

Fiecare pacient poate avea un numar diferit de temperaturi memorate

Reguli definitorii ale bazelor de date relationale

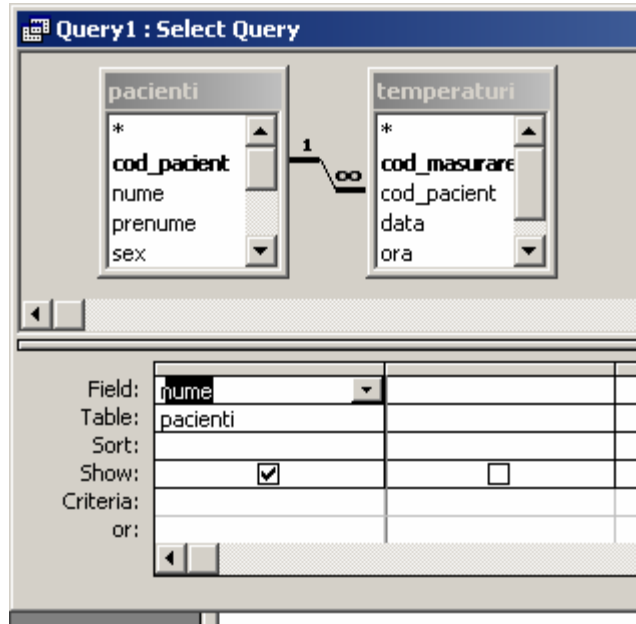
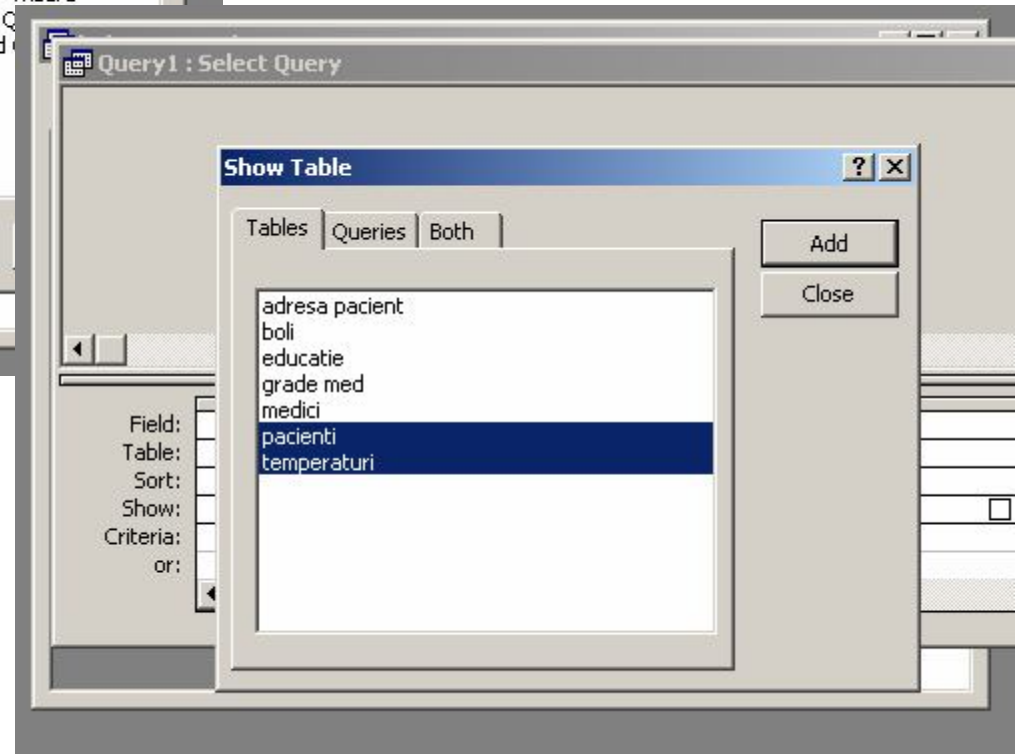
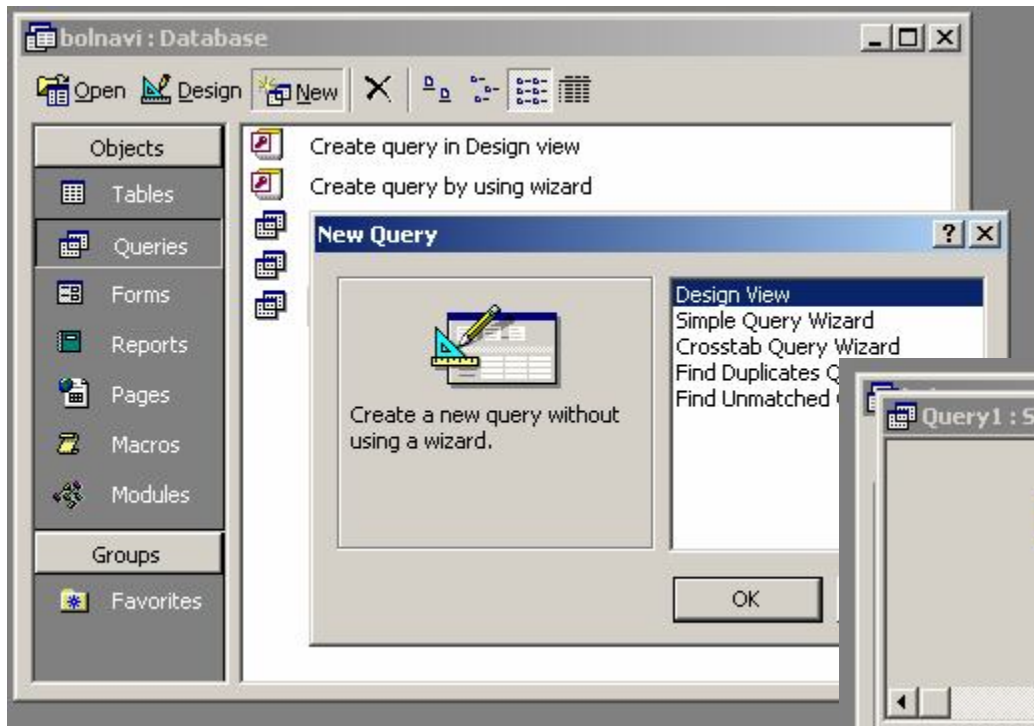
(dr. Codd 1969-1970)

- informațiile din baza de date sunt reprezentate exclusiv sub formă tabelară
- toate datele individuale dintr-un tabel sunt oricând accesibile prin specificarea numelui tabelului, a liniei și a coloanei
- baza de date poate include ca valide valorile nule
- baza de date reprezintă descrierea informațiilor înmagazinate într-un format logic simplificat de genul tabelelor
- modelul relațional are ca limbaj principal de interfațare SQL, însă poate suporta și alte limbaje
- vederile sunt actualizabile, dacă vederea curentă este un tabel

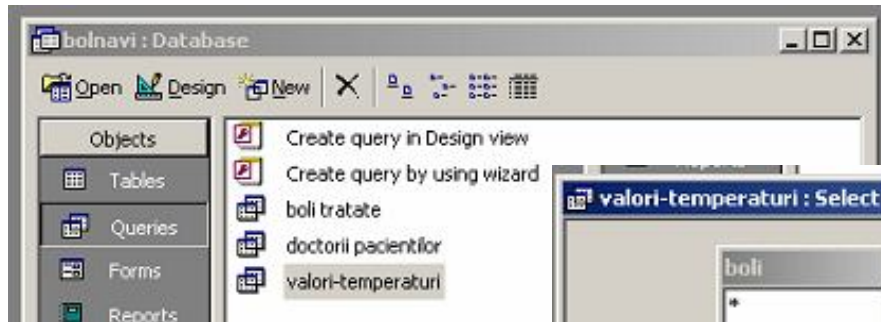
- modelul relațional tratează toate relațiile ca un singur operand pentru operațiile de actualizare, inserare și eliminare efectuate asupra datelor
- aspectele logice ale bazei de date sunt complet separate de aspectele fizice
- datele sunt conservate atunci când bazei de date i se aduc modificări ilogice
- regulile de integritate sunt create în SQL, fiind stocate în catalogul bazei de date și nu în aplicațiile individuale
- distribuția datelor (copierea datelor într-o bază de date aflată la distanță) către programele de aplicație are loc continuu
- regulile și restricțiile de integritate nu pot fi ocolite de nici un limbaj de acces
- sistemul manevrează bazele de date folosind exclusiv caracteristicile relaționalele

Queries

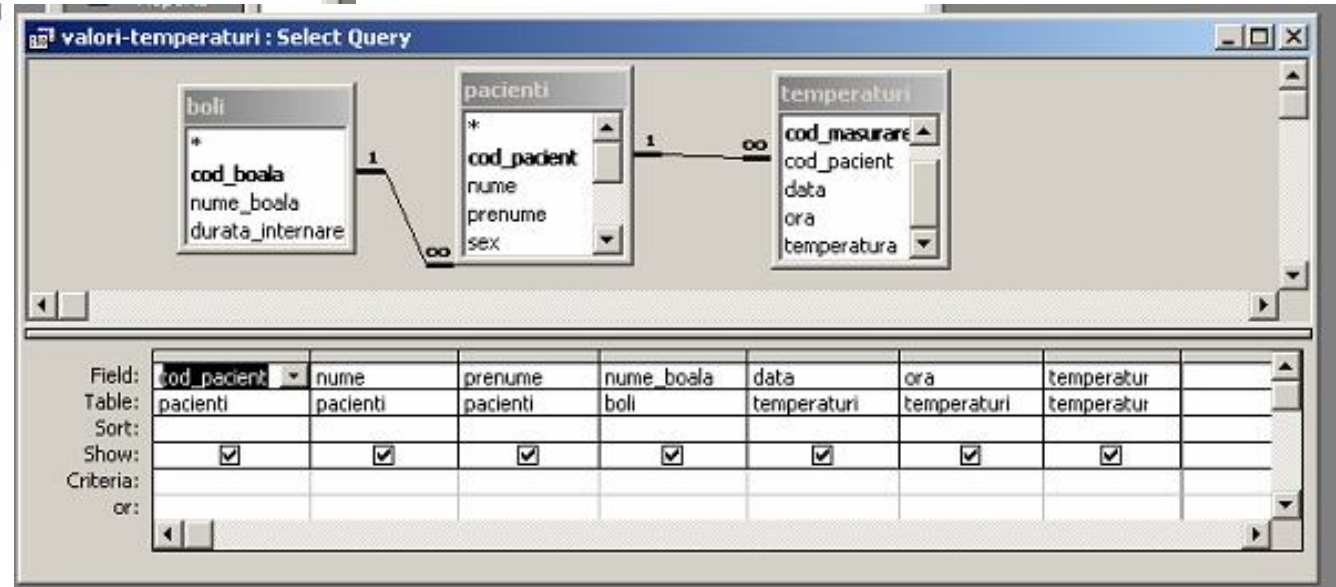
Interrogari



Queries - Interogari



Aflarea tuturor
temperaturilor
tuturor
pacientilor



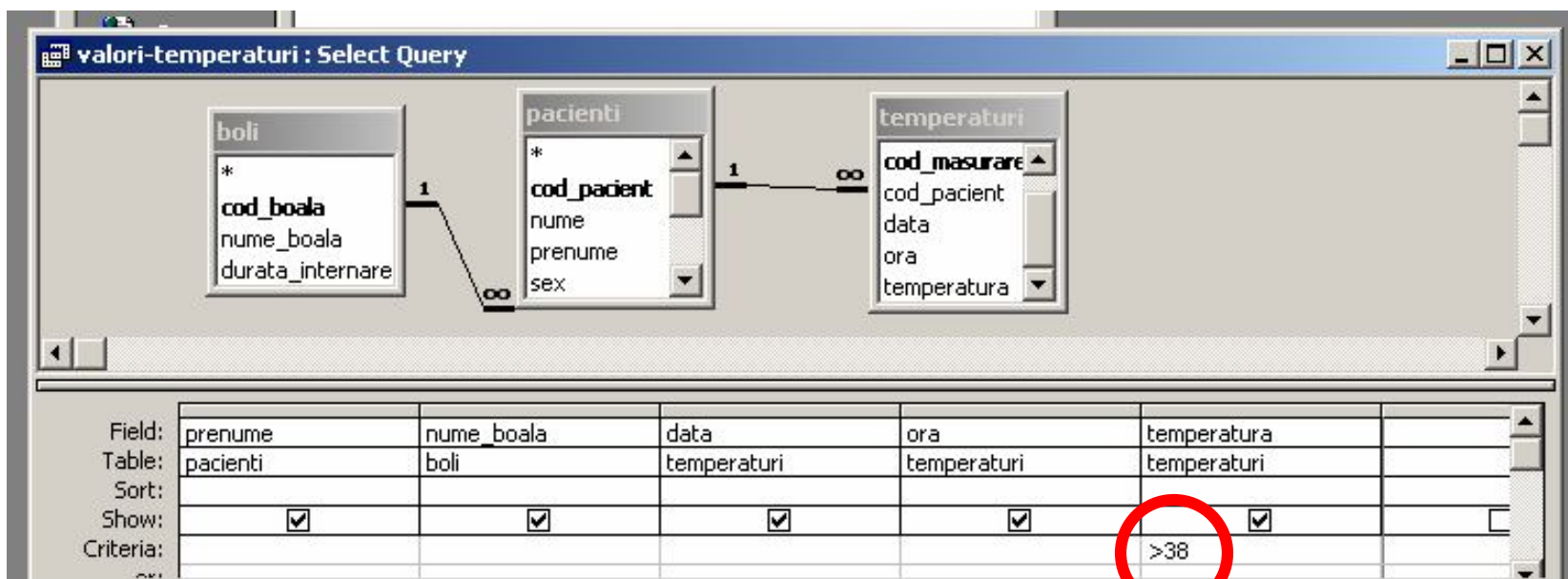
valori-temperaturi : Select Query

	cod_pacient	nume	prenume	nume_boala	data	ora	temperatura
▶	2	popescu	ion	boala1	1/2/2006	7:00	38
	2	popescu	ion	boala1	1/2/2006	15:00	38
	2	popescu	ion	boala1	1/2/2006	23:00	38.5
	2	popescu	ion	boala1	1/3/2006	7:00	38
	2	popescu	ion	boala1	1/3/2006	15:00	37.7
	2	popescu	ion	boala1	1/3/2006	23:00	37.5
	2	popescu	ion	boala1	1/4/2006	7:00	37.6
	2	popescu	ion	boala1	1/4/2006	15:00	37.5
	7	gica	ileana	boala5	1/1/2006	8:00	38.5
	7	gica	ileana	boala5	1/1/2006	12:00	39
	7	gica	ileana	boala5	1/1/2006	16:00	39.5

Record: 1 of 15

Queries Interrogari

Aflarea pacientilor febrili

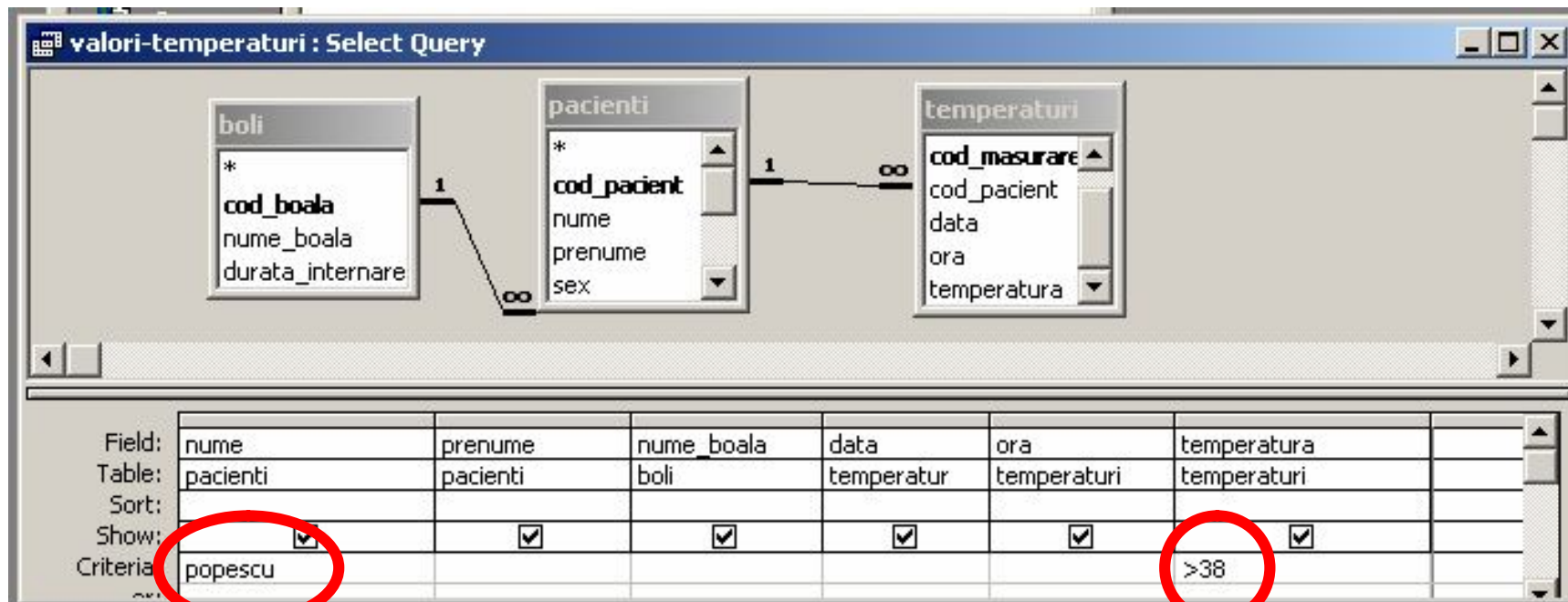


valori-temperaturi : Select Query

	cod_pacient	nume	prenume	nume_boala	data	ora	temperatura
▶	7	gica	ileana	boala5	1/1/2006	8:00	38.5
	7	gica	ileana	boala5	1/1/2006	12:00	39
	7	gica	ileana	boala5	1/1/2006	16:00	39.5
	7	gica	ileana	boala5	1/1/2006	20:00	40
	7	gica	ileana	boala5	1/1/2006	0:00	39.5
	7	gica	ileana	boala5	1/2/2006	4:00	39.5
	7	gica	ileana	boala5	1/2/2006	8:00	39
	2	popescu	ion	boala1	1/2/2006	23:00	38.5
*	(AutoNumber)						

Queries Interogari

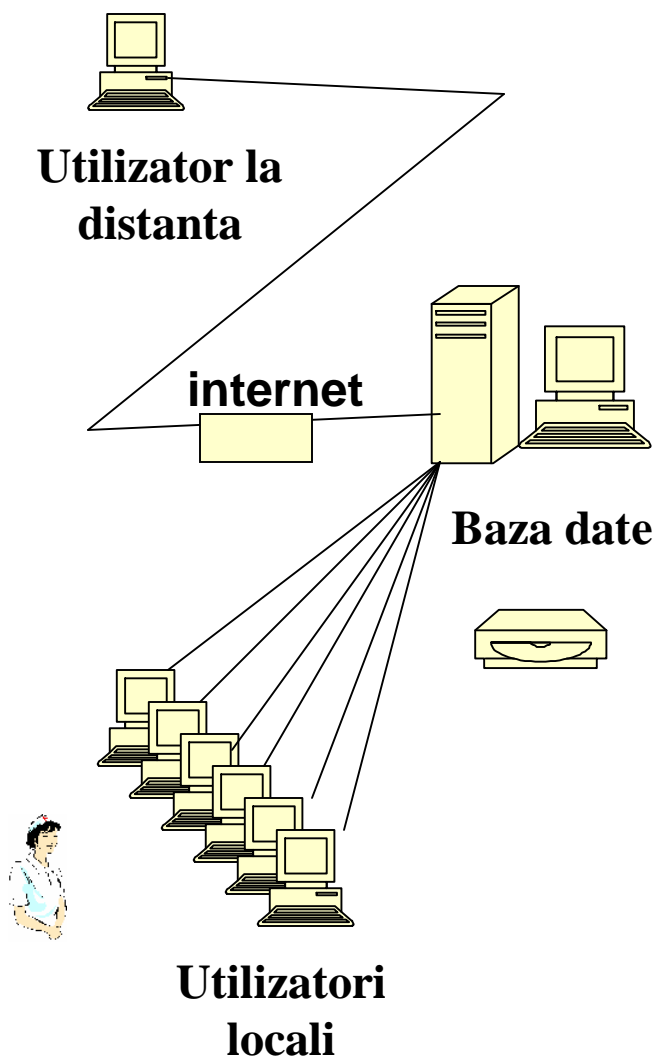
Gasirea momentelor febrile
pentru pacientul Popescu



Query Results (Field: nume, Table: pacienti):

	cod_pacient	nume	prenume	nume_boala	data	ora	temperatura
▶	2	popescu	ion	boala1	1/2/2006	23:00	38.5
*	(AutoNumber)						

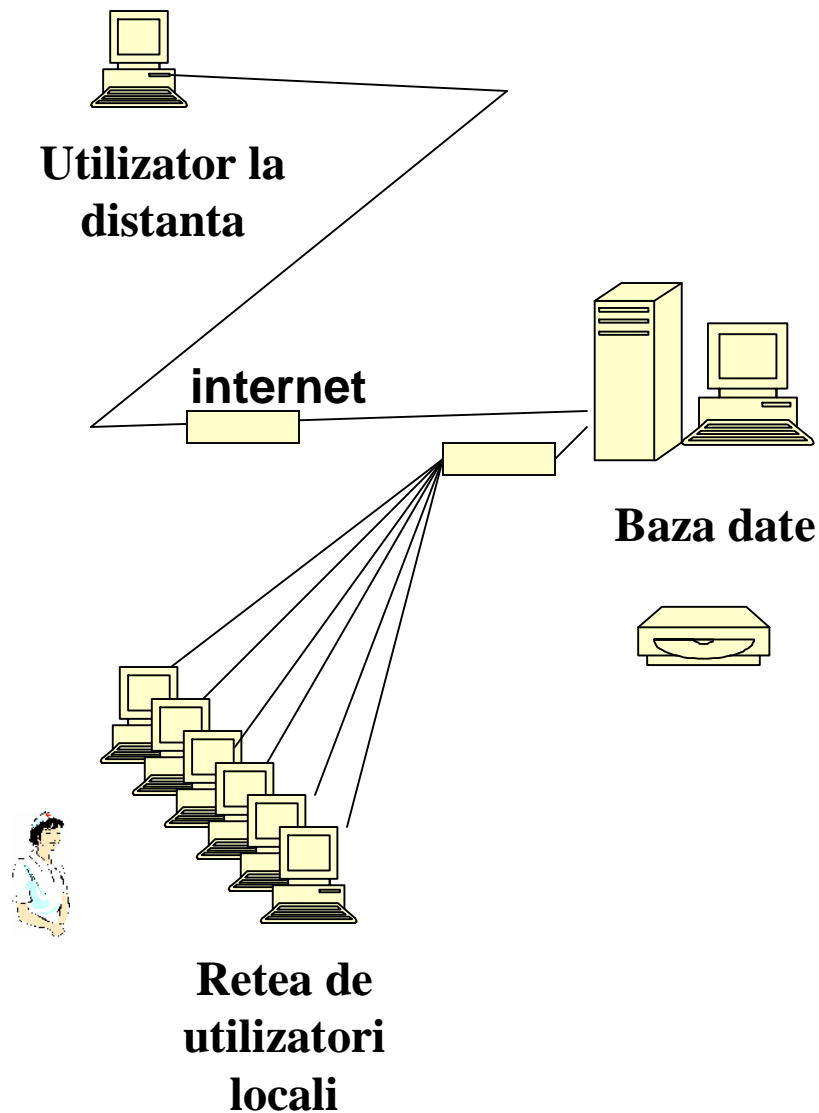
Arhitecturi de implementare



Modelul MAINFRAME-baza de date este organizată și stocată integral pe un sistem performant (denumit mainframe), de unde poate fi accesată de mai multe console utilizator prin intermediul unor aplicații de exploatare rezidente tot pe mainframe

- modelul s-a dovedit performant și sigur în exploatare, reușind să fie integrat în arhitecturi moderne
- aplicațiile au o structură rigidă, extinderea presupunând rezolvarea unor probleme privind organizarea și traficul de date

Arhitecturi de implementare



Modelul FILE-SERVER-

rețea centrată logic pe un calculator puternic, numit file server, de unde se alimentează cu date și aplicații celelalte calculatoare (PC) din rețea.

- file server-ul partajează datele pentru mai mulți utilizatori, oferindu-le sub forma unui disc din rețea
- pentru mărirea vitezei de acces la date este bine ca rețeaua să fie de tipul stea și nu liniară

Arhitecturi de implementare

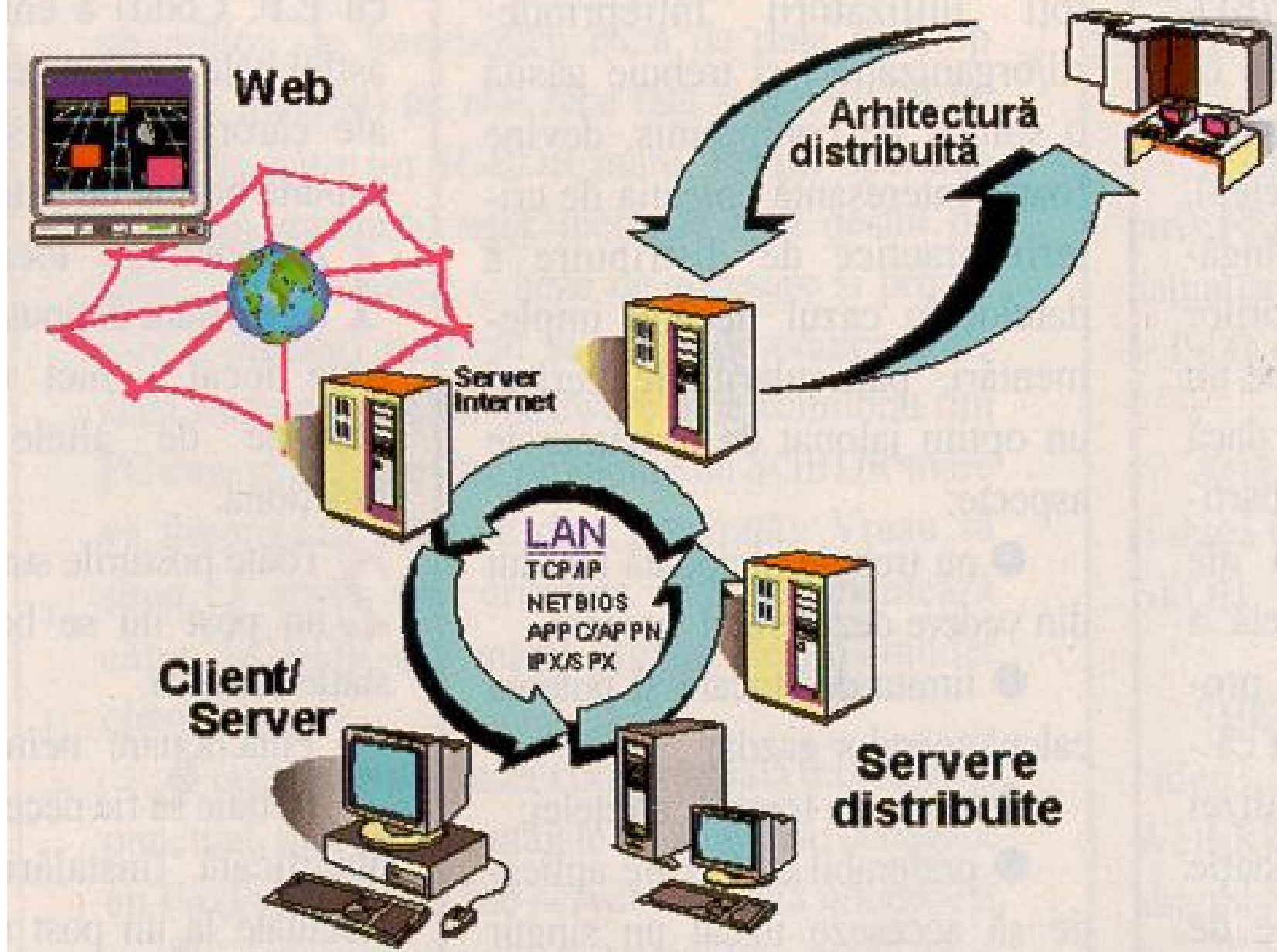
BAZE DE DATE DISTRIBUITE

- autonomia locală, datele locale sunt administrate local, nici un post nu depinde de altele pentru a lucra
- toate posturile sunt egale, nu există o stație centrală
- funcționarea este neîntreruptă, nu există opriri planificate
- transparența amplasării datelor, utilizatorii nu trebuie să știe unde sunt datele, pentru a le accesa
- transparența fragmentării datelor
- transparența duplicării datelor
- prelucrarea de interogări distribuite și actualizări distribuite
- independența de hardware și de sistemul de operare
- independența de bazele de date, datele pot proveni din SGBD diferite

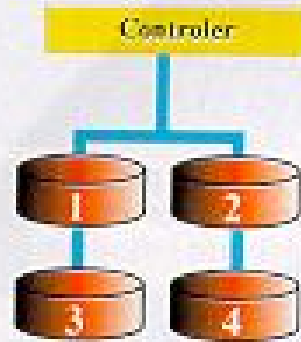
Modelul CLIENT-SERVER

- operare multiuser concurențială
- descongestionarea traficului prin rețea prin trimiterea doar a datelor țintă
- controlul drepturilor utilizatorilor și monitorizarea activității
- implementări unice de logică centralizată
- gestionarea tranzacțiilor client-server
- asigurarea de către server a integrității și consistenței datelor
- optimizarea organizării fizice a datelor
- recuperarea datelor în cazul căderii sistemului
- ridicarea performanțelor globale prin creșterea puterii serverului bazei de date

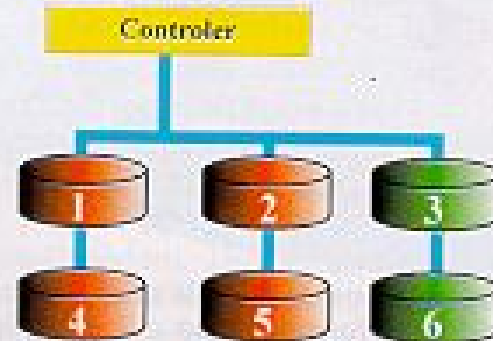
Baze de date în rețea eterogenă



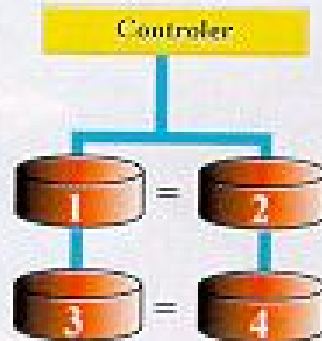
RAID 0



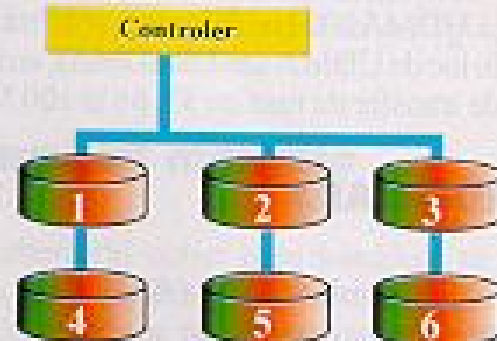
RAID 3



RAID 1



RAID 5



Exemple de configurații cu cele mai folosite nivele RAID. La RAID 0, datele sunt împărțite între cele patru HDD-uri ce formează o unitate (1 și 2, 3 și 4). La RAID 1, discurile 1 și 2 conțin aceeași informație (2 este oglinda lui 1), la fel și 3 și 4. La RAID 3, intervine informația de paritate, care se stochează pe un disc separat. La RAID 5, informația de paritate se stochează pe fiecare harddisk.